WHAT IS CLAIMED IS:

1. A method of parallel processing bit-synchronous HDLC data, comprising:

storing at least two bytes of bit-synchronous HDLC data in a shift register, wherein, at each successive clock cycle, a new incoming byte is shifted into said shift register and an old byte is shifted out of said shift register;

processing in parallel a plurality of bits within said shift register so as to detect a SOF sequence during a first clock cycle;

processing in parallel a plurality of bits within said shift register so as to detect an EOF sequence during at least one subsequent clock cycle; and

sending valid payload data bits to a packer logic unit, wherein said valid payload data bits comprise at least some bits shifted into said shift register between said SOF sequence and said EOF sequence.

2. The method of claim 1 further comprising processing in parallel a plurality of bits within said shift register to detect an Abort sequence during said at least one subsequent clock cycle, wherein if said Abort sequence is detected, all bits received after said SOF sequence are discarded and a search for a new SOF sequence is initiated.

3. The method of claim 1 further comprising:

processing in parallel a plurality of bits within said shift register to detect at least one stuff bit; and

discarding said at least one stuff bit, if detected, wherein said valid payload data bits comprise all bits shifted into said shift register between said SOF sequence and said EOF sequence, excluding said at least one stuff bit.

4. The method of claim 1 further comprising discarding all bits received between said SOF and EOF sequences if an amount of valid payload data bits received constitute less than a minimum number (N) of bytes.

5. The method of claim 1 wherein said acts of processing in parallel comprise providing a plurality of comparators coupled to said shift register, wherein each comparator is coupled to said shift register so as to parallel process a unique combination of eight successive bits contained within said shift register.

6. A method of parallel processing bit-synchronous data, comprising:

storing at least two bytes of bit-synchronous data in a shift register, wherein a newly received byte is shifted in and an old byte is shifted out of said shift register at each clock cycle;

processing in parallel a plurality of bits within said shift register so as to detect valid payload data bits; and

storing detected valid payload data bits in a packer logic unit for further processing.

7. The method of claim 6 wherein said act of processing in parallel comprises searching for a specified sequence of bits stored within said shift register during a first clock cycle.

8. The method of claim 7 wherein said act of searching for a specified sequence comprises searching for a start sequence within said shift register, wherein said valid payload data bits comprise at least some bits received after said start sequence.

9. The method of claim 8 further comprising searching for an end sequence within said shift register, after said start sequence has been detected, during at least one clock cycle subsequent to said first clock cycle, wherein said valid payload data bits comprise at least some bits received between said start and end sequences.

10. The method of claim 9 wherein said bit-synchronous data comprises bit-synchronous HDLC data, said start sequence comprises a SOF sequence and said end sequence comprises an EOF sequence.

11. The method of claim 10 further comprising processing in parallel a plurality of bits stored in said shift register to detect an Abort sequence during at least one clock cycle subsequent to said first clock cycle, wherein if said Abort sequence is detected, all bits received after said SOF sequence are discarded and a search for a new SOF sequence is initiated.

12. The method of claim 10 further comprising:

processing in parallel a plurality of bits within said shift register to detect at least one stuff bit; and

discarding said at least one stuff bit, if detected, wherein said valid payload data bits comprise all bits shifted into said shift register between said SOF sequence and said EOF sequence, excluding said at least one stuff bit.

13. The method of claim 10 further comprising discarding all bits received between said SOF and EOF sequences if an amount of valid payload data bits received constitute less than a minimum number (N) of bytes.

14. The method of claim 6 wherein said act of processing in parallel comprises providing a plurality of comparators coupled to said shift register, wherein each comparator is coupled to said shift register so as to parallel process a unique combination of eight successive bits contained within said shift register.

15. The method of claim 6 further comprising de-scrambling said at least two bytes prior to storing said at least two bytes in said shift register, wherein said de-scrambling comprises de-scrambling at least eight bits in parallel during a single clock cycle.

16. A system for parallel processing bit-synchronous data, comprising:

a shift register for storing a plurality of bits of bit-synchronous data, wherein a new plurality of bits is shifted in and an old plurality of bits is shifted out of said shift register during successive clock cycles; and

a de-framer unit, coupled to said shift register, for detecting valid payload data, wherein

said de-framer unit processes in parallel a plurality of bits within said shift register during a first clock cycle.

17. The system of claim 16 wherein said de-framing unit further processes in parallel said plurality of bits within said shift register to detect a specified sequence of bits during said first clock cycle.

18. The system of claim 17 wherein said de-framing unit searches for a start sequence within said shift register, wherein said valid payload data bits comprise at least some bits received after said start sequence.

19. The system of claim 18 wherein said de-framing unit further searches for an end sequence within said shift register, after said start sequence has been detected, during at least one clock cycle subsequent to said first clock cycle, wherein said valid payload data bits comprise at least some bits received between said start and end sequences.

20. The system of claim 16 wherein:

said bit-synchronous data comprises bit-synchronous HDLC data;

said shift register stores at least two bytes of said bit-synchronous HDLC data;

said new and old plurality of bits each comprise one byte of data;

said de-framer unit comprises a plurality of comparators wherein each comparator is coupled to a unique subset of eight successive bits contained within said shift register;

said start sequence comprises a SOF sequence;

said end sequence comprises an EOF sequence; and

said valid payload data bits comprise at least some bits received between said SOF and EOF sequences.

21. The system of claim 20 wherein said de-framer unit further processes in parallel a plurality of bits stored in said shift register to detect an Abort sequence during at least one clock cycle subsequent to said first clock cycle, wherein if said Abort sequence is detected, all bits

received after said SOF sequence are discarded and a search for a new SOF sequence is initiated.

22. The system of claim 20 wherein said de-framer unit further processes in parallel a plurality of bits within said shift register to detect at least one stuff bit and discards said at least one stuff bit, if detected, wherein said valid payload data bits comprise all bits shifted into said shift register between said SOF sequence and said EOF sequence, excluding said at least one stuff bit.

23. The system of claim 20 further comprising a packer logic unit, coupled to said de-framer unit, for storing said valid payload data received from said de-framer unit, said de-framer unit further comprising a look-up table (LUT) representing a plurality of bit sequences containing at least one stuff bit and wherein, if said at least one stuff bit is detected, said LUT passes through valid payload data bits to said packer logic unit while not allowing said at least one stuff bit to be sent to said packer logic unit.

24. The system of claim 23 wherein said packer logic unit discards said valid payload data received from said de-framer unit if an amount of valid payload data bits received constitute less than a minimum number (N) of bytes.

25. A bit-synchronous HDLC engine, comprising:

a shift register for storing at least two bytes of bit-synchronous HDLC data, wherein a new byte is shifted in and an old byte is shifted out of said shift register during each successive clock cycle; and

a de-framer unit, coupled to said shift register, for detecting valid payload data within said shift register, wherein said de-framer unit comprises a plurality of comparators for detecting a specified sequence of bits within said shift register during a first clock cycle, wherein each comparator is coupled to a unique subset of eight successive bits contained within said shift register.

26. The bit-synchronous HDLC engine of claim 25 wherein said de-framer unit detects a SOF sequence during said first clock cycle and detects an EOF sequence during a subsequent clock cycle, wherein said valid payload data comprises at least some of the bits received between said SOF sequence and said EOF sequence.

27. The bit-synchronous HDLC engine of claim 26 wherein said de-framer unit further processes in parallel a plurality of bits stored in said shift register to detect an Abort sequence during at least one clock cycle subsequent to said first clock cycle, wherein if said Abort sequence is detected, all bits received after said SOF sequence are discarded and a search for a new SOF sequence is initiated.

28. The bit-synchronous HDLC engine of claim 26 wherein said de-framer unit further processes in parallel a plurality of bits within said shift register to detect at least one stuff bit and discards said at least one stuff bit, if detected, wherein said valid payload data bits comprise all bits shifted into said shift register between said SOF sequence and said EOF sequence, excluding said at least one stuff bit.

29. The bit-synchronous HDLC engine of claim 28 further comprising a packer logic unit, coupled to said de-framer unit, for storing said valid payload data received from said de-framer unit, wherein said de-framer unit further comprises a look-up table (LUT) representing a plurality of bit sequences containing at least one stuff bit, wherein, if a stuff bit is detected, said LUT passes through valid payload data bits while not allowing said at least one stuff bit to be sent to said packer logic unit.

30. The bit-synchronous HDLC engine of claim 26 further comprising a packer logic unit, coupled to said de-framer unit, for storing said valid payload data received from said de-framer unit, wherein said packer logic unit discards said valid payload data received from said de-framer unit if an amount of valid payload data bits received constitute less than a minimum number (N) of bytes.

31. A system for parallel processing bit-synchronous data, comprising:

means for storing a plurality of bits of bit-synchronous data, wherein a new plurality of bits is stored and an old plurality of bits is expelled out of said means for storing during successive clock cycles; and

de-framer means, coupled to said means for storing, for detecting valid payload data and for processing in parallel a plurality of bits within said means for storing during a first clock cycle.

32. The system of claim 31 further comprising means for processing in parallel said plurality of bits within said means for storing to detect a specified sequence of bits during said first clock cycle.

33. The system of claim 32 wherein said means for processing in parallel comprises meand for detecting a start sequence contained within said means for storing, wherein said valid payload data bits comprise at least some bits received after said start sequence.

34. The system of claim 33 wherein said means for processing in parallel further comprising means for detecting an end sequence within said means for storing, after said start sequence has been detected, during at least one clock cycle subsequent to said first clock cycle, wherein said valid payload data bits comprise at least some bits received between said start and end sequences.

35. The system of claim 34 further comprising means for detecting an Abort sequence contained within said means for storing, during at least one clock cycle subsequent to said first clock cycle, wherein if said Abort sequence is detected, all bits received after said start sequence are discarded and a search for a new start sequence is initiated.

36. The system of claim 34 further comprising means for detecting at least one stuff bit contained within said means for storing and for discarding said at least one stuff bit, if detected, wherein said valid payload data bits comprise all bits stored in said means for storing after said start sequence and before said end sequence, excluding said at least one stuff bit.